Changes to the MIPP offline fluka and e907mc packages

Holger Meyer (WSU)

March 1, 2011

Abstract

Changes to several MIPP offline analysis packages[1] are described. These are *fluka*, *e907mc*, *E907MCInterface*. All modifications are fully backwards compatible.

1 Background and problem description

The simulation tool chain in MIPP data analysis consists of event generation (resulting in a stdhep file), tracking of events through the detector geometry in e907mc which is based on Geant3 (resulting in root data files with MC-truth and hit information) and digitization and reconstruction in an anamipp job using the offline framework (resulting in root data files with digits and tracks added and in root dst files). Event generation is done mainly with fluka and dpmjet. With dpmjet a primary interaction is generated and primary particles are written to the stdhep file. With fluka the primary interaction is tracked in fluka through the experimental target volume and tracks are written to the stdhep file as they cross the target volume boundary. The advantage is that all of fluka's interactions are taken into account inside the target. A disadvantage is that information on primary vertex position and multiplicity is not available in the later stages as it is not stored in the stdhep files. This hindered among other things the study of KNO scaling behavior in MIPP data.

It is also desirable to retain the information on what kind of interaction (elastic, inelastic, ...) created the particles in the event. This information is accessible in fluka and is preserved with the modifications presented here.

2 Problem solution overview

It was decided[2] to add the vertex information from fluka to the stdhep files while retaining the particles tracked by fluka through the target. In e907mc the particles from the target volume boundary crossing will be tracked as before (and output placed in folders "kine" and "hits" in the output root file) while additional information on the primary vertex particles is ignored and simply passed through into the output file into a new folder "kine-primvert". If vertex information is not available (as in the case of stdhep files before the modifications described here or for stdhep files from dpmjet), then e907mc behaves fully backwards compatible. Even if the *old* e907mc encounters an stdhep file prduced by the *new* fluka (a highly unusual, unnecessary use-case), despite the limitations in the stdhep file format, the new information in the stdhep files will be ignored because the old hep2geant routine skips over particels with stdhep status flags other than 1 or the MIPP beam status flag (usually 301) as they are considered to be decayed particles.

In the further offline analysis the changes are also completely transparent because the new information is put into a new folder that was simply not present in the old files and is thus ignored by the old software versions.

Fluka has been modified to write out the vertex particle information. The input file must specify a USERDUMP card in the fluka input file to activate this.

3 Problem solution details

3.1 fluka

Limits of stdhep files. Use status flag to encode particle number. The STDHEP manual[3] defines the ISTHEP integer as follows:

ISTHEP value	-	definition
0	-	null
1	-	final state particle
2	-	intermediate state
3	-	documentation line
4-10	-	reserved for future use
11-200	-	reserved for specific model use
201-...	-	reserved for users

To this was added before the MIPP convention:

ISTHEP value	-	definition
301	-	beam particle to be tracked backwards

Now the following codes are also used. Particles from the stdhep file with these codes must not be tracked in e907mc.

ISTHEP value	-	definition
390	-	particle produced at primary interaction
	-	(this particle or its daughters are also written out with ISTHEP code 1)
	-	beam particle, event primary interaction is....
	-	(see fluka manual p.361, USDRAW entry to MGDRAW routine)
399	-	no event, uninteracted beam particle
40x	-	interaction in fluka subroutine KASKAD (hadron and muon interactions)
400	-	elastic interaction
401	-	inelastic interaction
402	-	particle decay
403	-	delta ray generation
404	-	pair production
405	-	bremsstrahlung
410	-	radioactive decay
50x	-	interaction in fluka subroutine EMFSCO (electron, positron and photon interactions)
508	-	bremsstrahlung
510	-	Mller scattering
512	-	Bhabha scattering
514	-	in-flight annihilation
515	-	annihilation at rest
517	-	pair production
519	-	Compton scattering
521	-	photoelectric interaction
525	-	Rayleigh scattering interaction
60x	-	interaction in fluka subroutine KASNEU (low-energy neutron interactions)
600	-	neutron interaction
70x	-	interaction in fluka subroutine KASHEA (heavy ion interactions)
700	-	delta ray generation

Some of these codes could be argued to duplicate definitions provided in the standard. For example ISTHEP code 2 might be used for those primary particles that are decayed or interacted by fluka. However, it was deemed a cleaner implementation to consistently use codes in the range reserved for users. Of the ISTHEP codes for beam particles encoding the type of interaction only codes 399, 400, and 401 are expected to be used in MIPP. However, the other codes are defined here because fluka can issue these codes. They are the codes defined on page 361 of the fluka manual[4] with an offset of 300 added because the lower codes would otherwise conflict with other STDHEP ISTHEP codes.

The particles are stored in the event such that the beam track is stored first, followed by the primary vertex information, and then followed by the particles information at the target volume boundary. As particles are stored twice (or a particle and its daughters are stored, if the particle interacts inside the target), the total number of particles in the event (stdhep ntrk variable) is not the same as the number of particles in the interaction, the interaction multiplicity. However, the stdhep ntrk did not coincide with the interaction multiplicity before this change (which was the main reason for this modification). The multiplicity can now be determined from the stdhep data by looping over all particles in the event and counting those with ISTHEP status value of 390.

3.2 e907mc

The e907mc program structure as relevant here is as follows: For each event the routine hep2gean3 is called to get the stdhep particles copied into the Geant3 data structures. Then Geant3 routines track all the particles in Geant3 common blocks (and add secondary particles and hits). Finally the routines in the E907MCInterface package (MCIDoKine and MCIStoreHits) copy the information to the root data structures used in the MIPP offline software. Thus, in order to add the vertex particle information to the root output file without tracking these particles in Geant3, they will get handed to the E907MCInterface directly from hep2gean3 and never get put into common blocks or other data structures used by Geant3. The hep2gean3 routine has been modified in the past by MIPP to allow for the inverse beam flag and distribution of vertex positions throughout the target volume (for stdhep files from dpmjet).

3.3 E907MCInterface

- output into root file

3.4 DST

As we want this new information to be available in the DSTs the MIPPEventSummary and DSTMaker have been modified....

References

[1] For general information on MIPP and MIPP data analysis see `http://www-mipp.fnal.gov/` and documents linked there.

[2] The decision was based on discussion at weekly MIPP phone meetings. See H. Meyer, "Vertex information in stdhep files from fluka", MIPP-Doc-1062-v2

[3] Lynn Garren, "StdHep 5.06.01 Monte Carlo Standardization at FNAL Fortran and C Implementation" (20 November 2006), FNAL-CD-Doc-903-v15, page 5, available at `http://cd-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=903&version=15&filename=stdhep_50601_manual.pdf`

[4] Alfredo Ferrari, Paola R. Sala, Alberto Fassò, Johannes Ranft, "Fluka: a multi-particle transport code (Program version 2008)", available at `http://www.fluka.org/content/manuals/FM.pdf`